I hereby certify that this correspondence is being deposited with the U.S. Postal Service with sufficient postage as First Class Mail in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA , 22313-1450, on:

Date: 9/7/05 By: Susan Wall

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Appl. No.	**:**	**09/784,699**	**Confirmation No.**	**:**	**3149**
Applicant	**:**	**Benitez et al.**	**TC/A.U.**	**:**	**2157**
Filed	**:**	**February 14, 2001**			
Examiner	**:**	**Hussein A. El Chanti**			
Docket No.	**:**	**30126-8009.US01**	**Customer No.**	**:**	**22918**

Declaration of Prior Invention Under 37 C.F.R. § 1.131

Commissioner for Patents
P.O. Box 1450
Alexandria, VA

I. This Declaration establishes invention prior to September 26, 2000.

II. This Declaration is being made by Daniel T. Arai, i.e., a named inventor of the above-identified patent application.

III. Conception: Prior to September 26, 2000, we conceived the inventions currently presented in independent claims 1, 16, and 31, of the above-identified patent application. A list of these claims is attached hereto as Exhibit A. Claim 1 is exemplary of an embodiment of the inventions. Exhibit B includes a listing of files related to a product that is representative of the embodiment claimed in the exemplary independent claim 1. Exhibit B includes versions of software and documentation that were created in a Content Management System (CMS) prior to September 26, 2000. The dates of each file have been redacted.

Exhibit C includes a subset of content from the CMS as of September 25, 2000. The content is from files listed in Exhibit B, which are entitled Exhibit C1 through C18. Exhibit C correlates to the exemplary independent claim 1. These

correlations are for the purpose of example only, and not intended to limit the scope of the claims. TABLE 1 provides a rough correlation between Exhibit C and, for example, independent claim 1:

TABLE 1

EXHIBIT C (Examples only)	CLAIM 1
C2) • Functionality (pg 1) o The Client Network Interface (CNI) provides the interface for sending message to servers and provides threads for receiving responses and dispatching them appropriately. (para. 1). • Asynchronous Server Calls (pgs. 5 – 6) o The network send thread is periodically awoken, and it coalesces requests off the NW request queue and sends them to the server (pg. 6, para. 1). o The network receives thread waits for responses to come back from any server. (pg. 6, para. 2). o Finally, the response dispatch thread pulls responses off the response queue, and handles the work of dispatching them appropriately. (pg. 6, para. 3) C6) • Estream client network interface o Handles requests from Estream cache manager o Handles protocol interface to/from server C7) Diagram illustrating structure of server streaming of application programs across a computer network while executing application programs on an Estream client. C17) Abstract and descriptions of CORBA illustrate an implementation of a server framework for Estream. (pg. 1). C18) Diagram illustrating structure of server streaming of applications programs across a computer network while executing application programs on an Estream client.	(a) A process for intelligent server streaming of conventionally coded streamed application programs across a computer network while concurrently executing said streamed application programs on a client in a computer environment, comprising the steps of:

(pg. 7)	
C1) • AIMInstallApplication Prototype (pg. 16) o Step 5. Initializing the profile and prefetch data for this app. C5) • Installation of application o Contact designated App Server using id info, download meta-data about app, potentially including registry/DLL/filesys spoofing info, prefetching info, initial cache contents for app. o Perform initial installation & setup for app, after checking system for previously installed version of app & issuing any appropriate warnings. C10) • Functionality (pg. 1) o The AppInstallBlock is a block of code and data associated with a particular application. This AppInstallBlock contains the information needed by the Estream client to "initialize" the client machine before the Estream application is used for the first time. C11) Data format of the AppInstallBlock.	(b) downloading an initial portion of a streamed application program on said client wherein said streamed application program comprises page segments and wherein said initial portion of said streamed application remains on said client after terminating execution of said streamed application by said client;
C7) Estream client-server diagram comprising an application server. C8) • Finding an application server for a volume. (pg. 2). o The SLM will tell the client which application servers currently provide each volume. It may be necessary for the client to periodically poll the SLM to get up-to-date information about the state of the application servers.	(c) providing an application server;

C9)
• **Technical description of the invention. (pg. 12)**
  ○ **An application file server: Responds to requests by client application cache manager for portions of application's files and directory structure on the server. Transmits compressed information for better bandwidth utilization. (pg. 12, item #4).**
 	
C16)
• **Functionality (pg. 1)**
  ○ **The primary job of the App Server is to service client requests for application data blocks. (pg. 1, para. 4).**
  ○ **The App Server serves data derived from Estream Sets. (pg. 1, para. 5).**
 	
C18)
• **Application Server (pg. 26)**
  ○ **The application server is there to handle read requests for files accessed by Estream clients. Any file accessed on a client through the EFS can have this read request passed to an app server.**	

C7) Estream client-server diagram illustrating transmission of Estream sets.
 	
C9)
• **The process of building a new set of request replies for an application is called building an application stream set. (pg. 15, 3rd full paragraph).**
• **An application stream set contains (pg. 15, 3rd full paragraph):**
  ○ **A unique name of the application for reference purposes,**
  ○ **An index table used to quickly determine which reply to return for a given request,**
  ○ **The set of all possible request**
	(d) partitioning said streamed application program into said page segments on said application server;

replies. • The application stream set is built in the following manner.... (pg. 15, 4th full paragraph). C12) • The Estream Builder is a software program. It is used to convert locally installable applications into a data set suitable for streaming over the network. The streaming-enabled data set is called the Estream Set. This document describes the procedure used to convert locally installable applications into the Estream Set. (pg. 1, para. 1). C13) Estream Builder data flow diagram illustrating the conversion of locally installable applications into Estream sets. C14) • Functionality (pg. 1) o The Estream Application Builder Package Manager is responsible for packaging data gathered from the Installation Monitor, the Profile Manager, and the Upgrade Monitor into a set of data called the Estream set. (pg. 1, para. 1). C15) • Functionality (pg. 1) o The Estream set is a data set associated with an application suitable for streaming over the network. The Estream set is generated by the Estream Builder program. This program converts locally installable applications into the Estream set. This document describes the format of the Estream set. (pg. 1, para. 1). o Diagram illustrating format of the Estream set. (pg. 5).	
C7) Estream client-server diagram illustrating transmission of Estream sets to	(e) wherein said application server streams said page segments to said client upon said

Estream client. **C9)** • **From the point of view of the client and its operating system, the application is resident locally on the client; the execution controller negotiates with an appropriate server to allow the client to obtain (as needed) segments of the associated application files located on the servers. (pg. 12, para. 2).**	**client's request;**
C9) • **The execution controller is given an argument indicating which application is to be executed. From the point of view of the client and its operating system, the application is resident locally on the client; the execution controller negotiates with an appropriate server to allow the client to obtain (as needed) segments of the associated application files located on the servers. (pg. 12, para. 2).** • **If a server accepts the task of serving the application to the client, the execution controller passes the application access request on to the application remote file interface code. This code allows the client to reference file and directory information associated with the remote application as if it resided on a local physical disk device. (pg. 13, 3rd full paragraph).**	**(f) wherein the user starts said streamed application program as if said streamed application program were fully installed on said client;**

C5)
• **Execution of application (pg. 2)**
  ○ **Send unique certificate for application to appropriate ASP DRM Server, get back id for closest/best App Server & session id.**
  ○ **Contact designated App Server using id info, request file system data as necessary. Respond to running application's requests,**
	(g) wherein specific page segments are requested by said client's file system during execution of said streamed application program such that said streamed application program begins execution on said client prior to downloading all of said page segments; and

collect usage data. C9) • The client's operating system begins executing the requested application located remotely on a server. The operating system memory-maps the application and begins executing it, with the application remote file interface code obtaining control whenever the client system's page fault handler determines that the application's page is located on the remote disk drive. The page fault handler asks the application remote file interface code to place the appropriate page data in main memory. The application remote file interface code sends a request to the cache manger for the desired data. (pg. 13, 4th full paragraph). C12) • Data flow description (pg. 6) o The OS loads the application executable into memory and runs the executable. (pg. 7, step 12). o The executable file image is loaded into memory and starts executing. The application files will continuously be loaded into memory as needed. (pg. 7, step 13).	
C3) • Functionality (pg. 1) o The cache manager manages the on-disk cache of file system data, and the in-memory data structures for managing this cache. (pg. 1, para. 3). • Diagram illustrating overall client architecture comprising cache elements. (pg. 8). • Implementation of the cache manager. (pgs. 11 – 17). C4)	e) storing said page segments in a cache on said client.

• **Cache organization (pg. 1)**
  ○ **The cache will be contained in 2 or more files. One file will contain the cache indices, and one or more files will contain the data blocks for cached files.**
 	
C5)
• **Execution of application (pg. 2)**
  ○ **Contact designated App Server using id info, request file system data as necessary. Respond to running application's requests, collect usage data. Cache portions of applications, file system info, & user preference info.**	

IV. **Diligence:** **We diligently constructively reduced the invention to practice on Nov. 6, 2000. Attached, with dates redacted, as Exhibits D1 through D5 (collectively "Exhibit D") are exemplary documents produced between September 26, 2000 and constructive reduction to practice. These documents are in chronological order, and have redacted dates which occurred at irregular intervals but without interruption extending from our conception of the invention to our constructive reduction to practice of the invention. Exhibit D is as follow:**

a) **D1: Estream 1.0 planning document**

b) **D2: Estream server component framework low level design**

c) **D3: Estream set format low level design**

d) **D4: Estream 1.0 high level design**

e) **D5: Estream web server load monitoring applet low level design**

Exhibit D correlates to the exemplary independent claim 1. These correlations are for the purpose of example only, and not intended to limit the scope of the claims. TABLE 2 provides a rough correlation between Exhibit D and, for example, independent claim 1:

TABLE 2

EXHIBIT D (Examples only)	CLAIM 1
D1) Server group time estimates for implementation. D2) • Functionality (pg. 1) o The Server Component Framework provides a common basis on which server components are implemented. The framework provides a number of services such as common server initialization and configuration, messaging, state management, logging, and error handling. (pg. 1, para. 1). D5) • Functionality (pg. 1) o One of the requirements for the Estream web server is a facility for monitoring server load. Per this document, this facility will be provided by a graphical load-monitoring applet that will be available for deployment at customer sites as part of the Estream web server installation.	(a) A process for intelligent server streaming of conventionally coded streamed application programs across a computer network while concurrently executing said streamed application programs on a client in a computer environment, comprising the steps of:
D1) • Content o AppInstallBlk structure time estimate for implementation.	(b) downloading an initial portion of a streamed application program on said client wherein said streamed application program comprises page segments and wherein said initial portion of said streamed application remains on said client after terminating execution of said streamed application by said client;
D1) • Server group o App Server time estimate for implementation.	(c) providing an application server;
D1) • Content o File access monitor time estimate for implementation.	(d) partitioning said streamed application program into said page segments on said application server;

○ Packager time estimate for implementation. ○ Estream distribution time estimate for implementation. D3) • Functionality (pg. 1) ○ The Estream set is a data set associated with an application suitable for streaming over the network. • Estream set format diagram (pg. 6).	
D4) • App server ○ The application server is there to handle read requests for files accessed by Estream clients. (pg. 10).	(e) wherein said application server streams said page segments to said client upon said client's request;
D4) • Overview ○ A small client 'player" program to allow local execution of applications that reside on the servers. (pg. 1) ○ The user will now see standard shortcuts for subscribed applications, exactly as though the app were installed locally. (pg. 2, 6th box)	(f) wherein the user starts said streamed application program as if said streamed application program were fully installed on said client;
	(g) wherein specific page segments are requested by said client's file system during execution of said streamed application program such that said streamed application program begins execution on said client prior to downloading all of said page segments; and
D1) • Client group ○ Estream cache manager time estimate for implementation. D4) • Client components (pg. 7) ○ ECM: the Estream cache manager. This is the user-space component that handles requests from the Estream File System	e) storing said page segments in a cache on said client.

Driver, and manages the on-disk and in-memory cache of file contents.	

V. **We hereby declare that all statements made herein of our own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, (18 U.S.C. §1001) and that such willful false statements may jeopardize the validity of this application or any patent issued thereon.**

Daniel T. Arai

Daniel T. Arai

Date August 12, 2005

I hereby certify that this correspondence is being deposited with the U.S. Postal Service with sufficient postage as First Class Mail in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA , 22313-1450, on:

Date: 9/7/05 By: Susan V Bel

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Appl. No.	**:**	**09/784,699**	**Confirmation No.**	**:**	**3149**
Applicant	**:**	**Benitez et al.**	**TC/A.U.**	**:**	**2157**
Filed	**:**	**February 14, 2001**			
Examiner	**:**	**Hussein A. El Chanti**			
Docket No.	**:**	**30126-8009.US01**	**Customer No.**	**:**	**22918**

Declaration of Prior Invention Under 37 C.F.R. § 1.131

Commissioner for Patents
P.O. Box 1450
Alexandria, VA

I. This Declaration establishes invention prior to September 26, 2000.

II. This Declaration is being made by Manuel E. Benitez, i.e., a named inventor of the above-identified patent application.

III. Conception: Prior to September 26, 2000, we conceived the inventions currently presented in independent claims 1, 16, and 31, of the above-identified patent application. A list of these claims is attached hereto as Exhibit A. Claim 1 is exemplary of an embodiment of the inventions. Exhibit B includes a listing of files related to a product that is representative of the embodiment claimed in the exemplary independent claim 1. Exhibit B includes versions of software and documentation that were created in a Content Management System (CMS) prior to September 26, 2000. The dates of each file have been redacted.

Exhibit C includes a subset of content from the CMS as of September 25, 2000. The content is from files listed in Exhibit B, which are entitled Exhibit C1 through C18. Exhibit C correlates to the exemplary independent claim 1. These

correlations are for the purpose of example only, and not intended to limit the scope of the claims. TABLE 1 provides a rough correlation between Exhibit C and, for example, independent claim 1:

TABLE 1

EXHIBIT C (Examples only)	CLAIM 1
C2) • Functionality (pg 1) o The Client Network Interface (CNI) provides the interface for sending message to servers and provides threads for receiving responses and dispatching them appropriately. (para. 1). • Asynchronous Server Calls (pgs. 5 – 6) o The network send thread is periodically awoken, and it coalesces requests off the NW request queue and sends them to the server (pg. 6, para. 1). o The network receives thread waits for responses to come back from any server. (pg. 6, para. 2). o Finally, the response dispatch thread pulls responses off the response queue, and handles the work of dispatching them appropriately. (pg. 6, para. 3) C6) • Estream client network interface o Handles requests from Estream cache manager o Handles protocol interface to/from server C7) Diagram illustrating structure of server streaming of application programs across a computer network while executing application programs on an Estream client. C17) Abstract and descriptions of CORBA illustrate an implementation of a server framework for Estream. (pg. 1). C18) Diagram illustrating structure of server streaming of applications programs across a computer network while executing application programs on an Estream client.	(a) A process for intelligent server streaming of conventionally coded streamed application programs across a computer network while concurrently executing said streamed application programs on a client in a computer environment, comprising the steps of:

(pg. 7)	
C1) • AIMInstallApplication Prototype (pg. 16) o Step 5. Initializing the profile and prefetch data for this app. C5) • Installation of application o Contact designated App Server using id info, download meta-data about app, potentially including registry/DLL/filesys spoofing info, prefetching info, initial cache contents for app. o Perform initial installation & setup for app, after checking system for previously installed version of app & issuing any appropriate warnings. C10) • Functionality (pg. 1) o The AppInstallBlock is a block of code and data associated with a particular application. This AppInstallBlock contains the information needed by the Estream client to "initialize" the client machine before the Estream application is used for the first time. C11) Data format of the AppInstallBlock.	(b) downloading an initial portion of a streamed application program on said client wherein said streamed application program comprises page segments and wherein said initial portion of said streamed application remains on said client after terminating execution of said streamed application by said client;
C7) Estream client-server diagram comprising an application server. C8) • Finding an application server for a volume. (pg. 2). o The SLM will tell the client which application servers currently provide each volume. It may be necessary for the client to periodically poll the SLM to get up-to-date information about the state of the application servers.	(c) providing an application server;

C9)
• **Technical description of the invention. (pg. 12)**
  ○ **An application file server: Responds to requests by client application cache manager for portions of application's files and directory structure on the server. Transmits compressed information for better bandwidth utilization. (pg. 12, item #4).**
 	
C16)
• **Functionality (pg. 1)**
  ○ **The primary job of the App Server is to service client requests for application data blocks. (pg. 1, para. 4).**
  ○ **The App Server serves data derived from Estream Sets. (pg. 1, para. 5).**
 	
C18)
• **Application Server (pg. 26)**
  ○ **The application server is there to handle read requests for files accessed by Estream clients. Any file accessed on a client through the EFS can have this read request passed to an app server.**	

C7) Estream client-server diagram illustrating transmission of Estream sets.
 	
C9)
• **The process of building a new set of request replies for an application is called building an application stream set. (pg. 15, 3rd full paragraph).**
• **An application stream set contains (pg. 15, 3rd full paragraph):**
  ○ **A unique name of the application for reference purposes,**
  ○ **An index table used to quickly determine which reply to return for a given request,**
  ○ **The set of all possible request**
	(d) partitioning said streamed application program into said page segments on said application server;

replies. • The application stream set is built in the following manner.... (pg. 15, 4th full paragraph). C12) • The Estream Builder is a software program. It is used to convert locally installable applications into a data set suitable for streaming over the network. The streaming-enabled data set is called the Estream Set. This document describes the procedure used to convert locally installable applications into the Estream Set. (pg. 1, para. 1). C13) Estream Builder data flow diagram illustrating the conversion of locally installable applications into Estream sets. C14) • Functionality (pg. 1) o The Estream Application Builder Package Manager is responsible for packaging data gathered from the Installation Monitor, the Profile Manager, and the Upgrade Monitor into a set of data called the Estream set. (pg. 1, para. 1). C15) • Functionality (pg. 1) o The Estream set is a data set associated with an application suitable for streaming over the network. The Estream set is generated by the Estream Builder program. This program converts locally installable applications into the Estream set. This document describes the format of the Estream set. (pg. 1, para. 1). o Diagram illustrating format of the Estream set. (pg. 5).	
C7) Estream client-server diagram illustrating transmission of Estream sets to	(e) wherein said application server streams said page segments to said client upon said

Estream client. C9) • **From the point of view of the client and its operating system, the application is resident locally on the client; the execution controller negotiates with an appropriate server to allow the client to obtain (as needed) segments of the associated application files located on the servers. (pg. 12, para. 2).**	**client's request;**
C9) • **The execution controller is given an argument indicating which application is to be executed. From the point of view of the client and its operating system, the application is resident locally on the client; the execution controller negotiates with an appropriate server to allow the client to obtain (as needed) segments of the associated application files located on the servers. (pg. 12, para. 2).** • **If a server accepts the task of serving the application to the client, the execution controller passes the application access request on to the application remote file interface code. This code allows the client to reference file and directory information associated with the remote application as if it resided on a local physical disk device. (pg. 13, 3rd full paragraph).**	**(f) wherein the user starts said streamed application program as if said streamed application program were fully installed on said client;**
C5) • **Execution of application (pg. 2)** o **Send unique certificate for application to appropriate ASP DRM Server, get back id for closest/best App Server & session id.** o **Contact designated App Server using id info, request file system data as necessary. Respond to running application's requests,**	**(g) wherein specific page segments are requested by said client's file system during execution of said streamed application program such that said streamed application program begins execution on said client prior to downloading all of said page segments; and**

collect usage data. C9) • The client's operating system begins executing the requested application located remotely on a server. The operating system memory-maps the application and begins executing it, with the application remote file interface code obtaining control whenever the client system's page fault handler determines that the application's page is located on the remote disk drive. The page fault handler asks the application remote file interface code to place the appropriate page data in main memory. The application remote file interface code sends a request to the cache manger for the desired data. (pg. 13, 4th full paragraph). C12) • Data flow description (pg. 6) o The OS loads the application executable into memory and runs the executable. (pg. 7, step 12). o The executable file image is loaded into memory and starts executing. The application files will continuously be loaded into memory as needed. (pg. 7, step 13).	
C3) • Functionality (pg. 1) o The cache manager manages the on-disk cache of file system data, and the in-memory data structures for managing this cache. (pg. 1, para. 3). • Diagram illustrating overall client architecture comprising cache elements. (pg. 8). • Implementation of the cache manager. (pgs. 11 – 17). C4)	e) storing said page segments in a cache on said client.

• **Cache organization (pg. 1)**
  ○ **The cache will be contained in 2 or more files. One file will contain the cache indices, and one or more files will contain the data blocks for cached files.**
 	
C5)
• **Execution of application (pg. 2)**
  ○ **Contact designated App Server using id info, request file system data as necessary. Respond to running application's requests, collect usage data. Cache portions of applications, file system info, & user preference info.**	

IV. Diligence: We diligently constructively reduced the invention to practice on Nov. 6, 2000. Attached, with dates redacted, as Exhibits D1 through D5 (collectively "Exhibit D") are exemplary documents produced between September 26, 2000 and constructive reduction to practice. These documents are in chronological order, and have redacted dates which occurred at irregular intervals but without interruption extending from our conception of the invention to our constructive reduction to practice of the invention. Exhibit D is as follow:

a) D1: Estream 1.0 planning document

b) D2: Estream server component framework low level design

c) D3: Estream set format low level design

d) D4: Estream 1.0 high level design

e) D5: Estream web server load monitoring applet low level design

Exhibit D correlates to the exemplary independent claim 1. These correlations are for the purpose of example only, and not intended to limit the scope of the claims. TABLE 2 provides a rough correlation between Exhibit D and, for example, independent claim 1:

TABLE 2

EXHIBIT D (Examples only)	CLAIM 1
D1) Server group time estimates for implementation. D2) • Functionality (pg. 1) o The Server Component Framework provides a common basis on which server components are implemented. The framework provides a number of services such as common server initialization and configuration, messaging, state management, logging, and error handling. (pg. 1, para. 1). D5) • Functionality (pg. 1) o One of the requirements for the Estream web server is a facility for monitoring server load. Per this document, this facility will be provided by a graphical load-monitoring applet that will be available for deployment at customer sites as part of the Estream web server installation.	(a) A process for intelligent server streaming of conventionally coded streamed application programs across a computer network while concurrently executing said streamed application programs on a client in a computer environment, comprising the steps of:
D1) • Content o AppInstallBlk structure time estimate for implementation.	(b) downloading an initial portion of a streamed application program on said client wherein said streamed application program comprises page segments and wherein said initial portion of said streamed application remains on said client after terminating execution of said streamed application by said client;
D1) • Server group o App Server time estimate for implementation.	(c) providing an application server;
D1) • Content o File access monitor time estimate for implementation.	(d) partitioning said streamed application program into said page segments on said application server;

○ Packager time estimate for implementation. ○ Estream distribution time estimate for implementation. D3) • Functionality (pg. 1) ○ The Estream set is a data set associated with an application suitable for streaming over the network. • Estream set format diagram (pg. 6).	
D4) • App server ○ The application server is there to handle read requests for files accessed by Estream clients. (pg. 10).	(e) wherein said application server streams said page segments to said client upon said client's request;
D4) • Overview ○ A small client 'player" program to allow local execution of applications that reside on the servers. (pg. 1) ○ The user will now see standard shortcuts for subscribed applications, exactly as though the app were installed locally. (pg. 2, 6th box)	(f) wherein the user starts said streamed application program as if said streamed application program were fully installed on said client;
	(g) wherein specific page segments are requested by said client's file system during execution of said streamed application program such that said streamed application program begins execution on said client prior to downloading all of said page segments; and
D1) • Client group ○ Estream cache manager time estimate for implementation. D4) • Client components (pg. 7) ○ ECM: the Estream cache manager. This is the user-space component that handles requests from the Estream File System	e) storing said page segments in a cache on said client.

Driver, and manages the on-disk and in-memory cache of file contents.	

V. **We hereby declare that all statements made herein of our own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, (18 U.S.C. §1001) and that such willful false statements may jeopardize the validity of this application or any patent issued thereon.**

Manuel E. Benitez

Date August 25, 2005

I hereby certify that this correspondence is being deposited with the U.S. Postal Service with sufficient postage as First Class Mail in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA , 22313-1450, on:

Date: 9/7/05 By: _____

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Appl. No.	:	**09/784,699**	**Confirmation No.**	:	**3149**
Applicant	:	**Benitez et al.**	**TC/A.U.**	:	**2157**
Filed	:	**February 14, 2001**			
Examiner	:	**Hussein A. El Chanti**			
Docket No.	:	**30126-8009.US01**	**Customer No.**	:	**22918**

Declaration of Prior Invention Under 37 C.F.R. § 1.131

Commissioner for Patents
P.O. Box 1450
Alexandria, VA

I. This Declaration establishes invention prior to September 26, 2000.

II. This Declaration is being made by Anne M. Holler, i.e., a named inventor of the above-identified patent application.

III. Conception: Prior to September 26, 2000, we conceived the inventions currently presented in independent claims 1, 16, and 31, of the above-identified patent application. A list of these claims is attached hereto as Exhibit A. Claim 1 is exemplary of an embodiment of the inventions. Exhibit B includes a listing of files related to a product that is representative of the embodiment claimed in the exemplary independent claim 1. Exhibit B includes versions of software and documentation that were created in a Content Management System (CMS) prior to September 26, 2000. The dates of each file have been redacted.

Exhibit C includes a subset of content from the CMS as of September 25, 2000. The content is from files listed in Exhibit B, which are entitled Exhibit C1 through C18. Exhibit C correlates to the exemplary independent claim 1. These

correlations are for the purpose of example only, and not intended to limit the scope of the claims. TABLE 1 provides a rough correlation between Exhibit C and, for example, independent claim 1:

TABLE 1

EXHIBIT C (Examples only)	CLAIM 1
C2) • Functionality (pg 1) o The Client Network Interface (CNI) provides the interface for sending message to servers and provides threads for receiving responses and dispatching them appropriately. (para. 1). • Asynchronous Server Calls (pgs. 5 – 6) o The network send thread is periodically awoken, and it coalesces requests off the NW request queue and sends them to the server (pg. 6, para. 1). o The network receives thread waits for responses to come back from any server. (pg. 6, para. 2). o Finally, the response dispatch thread pulls responses off the response queue, and handles the work of dispatching them appropriately. (pg. 6, para. 3) C6) • Estream client network interface o Handles requests from Estream cache manager o Handles protocol interface to/from server C7) Diagram illustrating structure of server streaming of application programs across a computer network while executing application programs on an Estream client. C17) Abstract and descriptions of CORBA illustrate an implementation of a server framework for Estream. (pg. 1). C18) Diagram illustrating structure of server streaming of applications programs across a computer network while executing application programs on an Estream client.	(a) A process for intelligent server streaming of conventionally coded streamed application programs across a computer network while concurrently executing said streamed application programs on a client in a computer environment, comprising the steps of:

(pg. 7) C1) • AIMInstallApplication Prototype (pg. 16) o Step 5. Initializing the profile and prefetch data for this app. C5) • Installation of application o Contact designated App Server using id info, download meta-data about app, potentially including registry/DLL/filesys spoofing info, prefetching info, initial cache contents for app. o Perform initial installation & setup for app, after checking system for previously installed version of app & issuing any appropriate warnings. C10) • Functionality (pg. 1) o The AppInstallBlock is a block of code and data associated with a particular application. This AppInstallBlock contains the information needed by the Estream client to "initialize" the client machine before the Estream application is used for the first time. C11) Data format of the AppInstallBlock.	(b) downloading an initial portion of a streamed application program on said client wherein said streamed application program comprises page segments and wherein said initial portion of said streamed application remains on said client after terminating execution of said streamed application by said client;
C7) Estream client-server diagram comprising an application server. C8) • Finding an application server for a volume. (pg. 2). o The SLM will tell the client which application servers currently provide each volume. It may be necessary for the client to periodically poll the SLM to get up-to-date information about the state of the application servers.	(c) providing an application server;

C9)
• **Technical description of the invention. (pg. 12)**
  ○ **An application file server: Responds to requests by client application cache manager for portions of application's files and directory structure on the server. Transmits compressed information for better bandwidth utilization. (pg. 12, item #4).**
 	
C16)
• **Functionality (pg. 1)**
  ○ **The primary job of the App Server is to service client requests for application data blocks. (pg. 1, para. 4).**
  ○ **The App Server serves data derived from Estream Sets. (pg. 1, para. 5).**
 	
C18)
• **Application Server (pg. 26)**
  ○ **The application server is there to handle read requests for files accessed by Estream clients. Any file accessed on a client through the EFS can have this read request passed to an app server.**	

C7) Estream client-server diagram illustrating transmission of Estream sets.
 	
C9)
• **The process of building a new set of request replies for an application is called building an application stream set. (pg. 15, 3rd full paragraph).**
• **An application stream set contains (pg. 15, 3rd full paragraph):**
  ○ **A unique name of the application for reference purposes,**
  ○ **An index table used to quickly determine which reply to return for a given request,**
  ○ **The set of all possible request**
	(d) partitioning said streamed application program into said page segments on said application server;

replies. • The application stream set is built in the following manner.... (pg. 15, 4th full paragraph). C12) • The Estream Builder is a software program. It is used to convert locally installable applications into a data set suitable for streaming over the network. The streaming-enabled data set is called the Estream Set. This document describes the procedure used to convert locally installable applications into the Estream Set. (pg. 1, para. 1). C13) Estream Builder data flow diagram illustrating the conversion of locally installable applications into Estream sets. C14) • Functionality (pg. 1) o The Estream Application Builder Package Manager is responsible for packaging data gathered from the Installation Monitor, the Profile Manager, and the Upgrade Monitor into a set of data called the Estream set. (pg. 1, para. 1). C15) • Functionality (pg. 1) o The Estream set is a data set associated with an application suitable for streaming over the network. The Estream set is generated by the Estream Builder program. This program converts locally installable applications into the Estream set. This document describes the format of the Estream set. (pg. 1, para. 1). o Diagram illustrating format of the Estream set. (pg. 5).	
C7) Estream client-server diagram illustrating transmission of Estream sets to	(e) wherein said application server streams said page segments to said client upon said

Estream client. **C9)** • **From the point of view of the client and its operating system, the application is resident locally on the client; the execution controller negotiates with an appropriate server to allow the client to obtain (as needed) segments of the associated application files located on the servers. (pg. 12, para. 2).**	**client's request;**
C9) • **The execution controller is given an argument indicating which application is to be executed. From the point of view of the client and its operating system, the application is resident locally on the client; the execution controller negotiates with an appropriate server to allow the client to obtain (as needed) segments of the associated application files located on the servers. (pg. 12, para. 2).** • **If a server accepts the task of serving the application to the client, the execution controller passes the application access request on to the application remote file interface code. This code allows the client to reference file and directory information associated with the remote application as if it resided on a local physical disk device. (pg. 13, 3rd full paragraph).**	**(f) wherein the user starts said streamed application program as if said streamed application program were fully installed on said client;**

C5)
• **Execution of application (pg. 2)**
  ○ **Send unique certificate for application to appropriate ASP DRM Server, get back id for closest/best App Server & session id.**
  ○ **Contact designated App Server using id info, request file system data as necessary. Respond to running application's requests,**
	(g) wherein specific page segments are requested by said client's file system during execution of said streamed application program such that said streamed application program begins execution on said client prior to downloading all of said page segments; and

collect usage data. C9) • The client's operating system begins executing the requested application located remotely on a server. The operating system memory-maps the application and begins executing it, with the application remote file interface code obtaining control whenever the client system's page fault handler determines that the application's page is located on the remote disk drive. The page fault handler asks the application remote file interface code to place the appropriate page data in main memory. The application remote file interface code sends a request to the cache manger for the desired data. (pg. 13, 4th full paragraph). C12) • Data flow description (pg. 6) o The OS loads the application executable into memory and runs the executable. (pg. 7, step 12). o The executable file image is loaded into memory and starts executing. The application files will continuously be loaded into memory as needed. (pg. 7, step 13).	
C3) • Functionality (pg. 1) o The cache manager manages the on-disk cache of file system data, and the in-memory data structures for managing this cache. (pg. 1, para. 3). • Diagram illustrating overall client architecture comprising cache elements. (pg. 8). • Implementation of the cache manager. (pgs. 11 – 17). C4)	e) storing said page segments in a cache on said client.

• **Cache organization (pg. 1)**
  ○ **The cache will be contained in 2 or more files. One file will contain the cache indices, and one or more files will contain the data blocks for cached files.**
 	
C5)
• **Execution of application (pg. 2)**
  ○ **Contact designated App Server using id info, request file system data as necessary. Respond to running application's requests, collect usage data. Cache portions of applications, file system info, & user preference info.**	

IV. Diligence: We diligently constructively reduced the invention to practice on Nov. 6, 2000. Attached, with dates redacted, as Exhibits D1 through D5 (collectively "Exhibit D") are exemplary documents produced between September 26, 2000 and constructive reduction to practice. These documents are in chronological order, and have redacted dates which occurred at irregular intervals but without interruption extending from our conception of the invention to our constructive reduction to practice of the invention. Exhibit D is as follow:

a) D1: Estream 1.0 planning document

b) D2: Estream server component framework low level design

c) D3: Estream set format low level design

d) D4: Estream 1.0 high level design

e) D5: Estream web server load monitoring applet low level design

Exhibit D correlates to the exemplary independent claim 1. These correlations are for the purpose of example only, and not intended to limit the scope of the claims. TABLE 2 provides a rough correlation between Exhibit D and, for example, independent claim 1:

TABLE 2

EXHIBIT D (Examples only)	CLAIM 1

D1) Server group time estimates for implementation.
 	
D2)
• **Functionality (pg. 1)**
  ○ **The Server Component Framework provides a common basis on which server components are implemented. The framework provides a number of services such as common server initialization and configuration, messaging, state management, logging, and error handling. (pg. 1, para. 1).**
 	
D5)
• **Functionality (pg. 1)**
  ○ **One of the requirements for the Estream web server is a facility for monitoring server load. Per this document, this facility will be provided by a graphical load-monitoring applet that will be available for deployment at customer sites as part of the Estream web server installation.**
	(a) A process for intelligent server streaming of conventionally coded streamed application programs across a computer network while concurrently executing said streamed application programs on a client in a computer environment, comprising the steps of:

D1) • **Content** ○ **AppInstallBlk structure time estimate for implementation.**	**(b) downloading an initial portion of a streamed application program on said client wherein said streamed application program comprises page segments and wherein said initial portion of said streamed application remains on said client after terminating execution of said streamed application by said client;**
D1) • **Server group** ○ **App Server time estimate for implementation.**	**(c) providing an application server;**
D1) • **Content** ○ **File access monitor time estimate for implementation.**	**(d) partitioning said streamed application program into said page segments on said application server;**

○ Packager time estimate for implementation. ○ Estream distribution time estimate for implementation. D3) • Functionality (pg. 1) ○ The Estream set is a data set associated with an application suitable for streaming over the network. • Estream set format diagram (pg. 6).	
D4) • App server ○ The application server is there to handle read requests for files accessed by Estream clients. (pg. 10).	(e) wherein said application server streams said page segments to said client upon said client's request;
D4) • Overview ○ A small client 'player" program to allow local execution of applications that reside on the servers. (pg. 1) ○ The user will now see standard shortcuts for subscribed applications, exactly as though the app were installed locally. (pg. 2, 6th box)	(f) wherein the user starts said streamed application program as if said streamed application program were fully installed on said client;
	(g) wherein specific page segments are requested by said client's file system during execution of said streamed application program such that said streamed application program begins execution on said client prior to downloading all of said page segments; and
D1) • Client group ○ Estream cache manager time estimate for implementation. D4) • Client components (pg. 7) ○ ECM: the Estream cache manager. This is the user-space component that handles requests from the Estream File System	e) storing said page segments in a cache on said client.

Driver, and manages the on-disk and in-memory cache of file contents.	

V. We hereby declare that all statements made herein of our own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, (18 U.S.C. §1001) and that such willful false statements may jeopardize the validity of this application or any patent issued thereon.

Anne M. Holler

Date August 24, 2005

I hereby certify that this correspondence is being deposited with the U.S. Postal Service with sufficient postage as First Class Mail in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA , 22313-1450, on:

Date: 9/7/05 By: Susan Hall

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Appl. No.	**:**	**09/784,699**	**Confirmation No.**	**:**	**3149**
Applicant	**:**	**Benitez et al.**	**TC/A.U.**	**:**	**2157**
Filed	**:**	**February 14, 2001**			
Examiner	**:**	**Hussein A. El Chanti**			
Docket No.	**:**	**30126-8009.US01**	**Customer No.**	**:**	**22918**

Declaration of Prior Invention Under 37 C.F.R. § 1.131

Commissioner for Patents
P.O. Box 1450
Alexandria, VA

I. This Declaration establishes invention prior to September 26, 2000.

II. This Declaration is being made by Lacky Shah, i.e., a named inventor of the above-identified patent application.

III. Conception: Prior to September 26, 2000, we conceived the inventions currently presented in independent claims 1, 16, and 31, of the above-identified patent application. A list of these claims is attached hereto as Exhibit A. Claim 1 is exemplary of an embodiment of the inventions. Exhibit B includes a listing of files related to a product that is representative of the embodiment claimed in the exemplary independent claim 1. Exhibit B includes versions of software and documentation that were created in a Content Management System (CMS) prior to September 26, 2000. The dates of each file have been redacted.

Exhibit C includes a subset of content from the CMS as of September 25, 2000. The content is from files listed in Exhibit B, which are entitled Exhibit C1 through C18. Exhibit C correlates to the exemplary independent claim 1. These

correlations are for the purpose of example only, and not intended to limit the scope of the claims. TABLE 1 provides a rough correlation between Exhibit C and, for example, independent claim 1:

TABLE 1

EXHIBIT C (Examples only)	CLAIM 1
C2) • Functionality (pg 1) o The Client Network Interface (CNI) provides the interface for sending message to servers and provides threads for receiving responses and dispatching them appropriately. (para. 1). • Asynchronous Server Calls (pgs. 5 – 6) o The network send thread is periodically awoken, and it coalesces requests off the NW request queue and sends them to the server (pg. 6, para. 1). o The network receives thread waits for responses to come back from any server. (pg. 6, para. 2). o Finally, the response dispatch thread pulls responses off the response queue, and handles the work of dispatching them appropriately. (pg. 6, para. 3) C6) • Estream client network interface o Handles requests from Estream cache manager o Handles protocol interface to/from server C7) Diagram illustrating structure of server streaming of application programs across a computer network while executing application programs on an Estream client. C17) Abstract and descriptions of CORBA illustrate an implementation of a server framework for Estream. (pg. 1). C18) Diagram illustrating structure of server streaming of applications programs across a computer network while executing application programs on an Estream client.	(a) A process for intelligent server streaming of conventionally coded streamed application programs across a computer network while concurrently executing said streamed application programs on a client in a computer environment, comprising the steps of:

(pg. 7)	
C1) • AIMInstallApplication Prototype (pg. 16) o Step 5. Initializing the profile and prefetch data for this app. C5) • Installation of application o Contact designated App Server using id info, download meta-data about app, potentially including registry/DLL/filesys spoofing info, prefetching info, initial cache contents for app. o Perform initial installation & setup for app, after checking system for previously installed version of app & issuing any appropriate warnings. C10) • Functionality (pg. 1) o The AppInstallBlock is a block of code and data associated with a particular application. This AppInstallBlock contains the information needed by the Estream client to "initialize" the client machine before the Estream application is used for the first time. C11) Data format of the AppInstallBlock.	(b) downloading an initial portion of a streamed application program on said client wherein said streamed application program comprises page segments and wherein said initial portion of said streamed application remains on said client after terminating execution of said streamed application by said client;
C7) Estream client-server diagram comprising an application server. C8) • Finding an application server for a volume. (pg. 2). o The SLM will tell the client which application servers currently provide each volume. It may be necessary for the client to periodically poll the SLM to get up-to-date information about the state of the application servers.	(c) providing an application server;

C9)
• **Technical description of the invention. (pg. 12)**
  ○ **An application file server: Responds to requests by client application cache manager for portions of application's files and directory structure on the server. Transmits compressed information for better bandwidth utilization. (pg. 12, item #4).**
 	
C16)
• **Functionality (pg. 1)**
  ○ **The primary job of the App Server is to service client requests for application data blocks. (pg. 1, para. 4).**
  ○ **The App Server serves data derived from Estream Sets. (pg. 1, para. 5).**
 	
C18)
• **Application Server (pg. 26)**
  ○ **The application server is there to handle read requests for files accessed by Estream clients. Any file accessed on a client through the EFS can have this read request passed to an app server.**	

C7) Estream client-server diagram illustrating transmission of Estream sets.
 	
C9)
• **The process of building a new set of request replies for an application is called building an application stream set. (pg. 15, 3rd full paragraph).**
• **An application stream set contains (pg. 15, 3rd full paragraph):**
  ○ **A unique name of the application for reference purposes,**
  ○ **An index table used to quickly determine which reply to return for a given request,**
  ○ **The set of all possible request**
	(d) partitioning said streamed application program into said page segments on said application server;

replies. • The application stream set is built in the following manner.... (pg. 15, 4th full paragraph). C12) • The Estream Builder is a software program. It is used to convert locally installable applications into a data set suitable for streaming over the network. The streaming-enabled data set is called the Estream Set. This document describes the procedure used to convert locally installable applications into the Estream Set. (pg. 1, para. 1). C13) Estream Builder data flow diagram illustrating the conversion of locally installable applications into Estream sets. C14) • Functionality (pg. 1) o The Estream Application Builder Package Manager is responsible for packaging data gathered from the Installation Monitor, the Profile Manager, and the Upgrade Monitor into a set of data called the Estream set. (pg. 1, para. 1). C15) • Functionality (pg. 1) o The Estream set is a data set associated with an application suitable for streaming over the network. The Estream set is generated by the Estream Builder program. This program converts locally installable applications into the Estream set. This document describes the format of the Estream set. (pg. 1, para. 1). o Diagram illustrating format of the Estream set. (pg. 5).	
C7) Estream client-server diagram illustrating transmission of Estream sets to	(e) wherein said application server streams said page segments to said client upon said

Estream client. **C9)** • **From the point of view of the client and its operating system, the application is resident locally on the client; the execution controller negotiates with an appropriate server to allow the client to obtain (as needed) segments of the associated application files located on the servers. (pg. 12, para. 2).**	**client's request;**
C9) • **The execution controller is given an argument indicating which application is to be executed. From the point of view of the client and its operating system, the application is resident locally on the client; the execution controller negotiates with an appropriate server to allow the client to obtain (as needed) segments of the associated application files located on the servers. (pg. 12, para. 2).** • **If a server accepts the task of serving the application to the client, the execution controller passes the application access request on to the application remote file interface code. This code allows the client to reference file and directory information associated with the remote application as if it resided on a local physical disk device. (pg. 13, 3rd full paragraph).**	**(f) wherein the user starts said streamed application program as if said streamed application program were fully installed on said client;**

C5)
• **Execution of application (pg. 2)**
  ○ **Send unique certificate for application to appropriate ASP DRM Server, get back id for closest/best App Server & session id.**
  ○ **Contact designated App Server using id info, request file system data as necessary. Respond to running application's requests,**
	(g) wherein specific page segments are requested by said client's file system during execution of said streamed application program such that said streamed application program begins execution on said client prior to downloading all of said page segments; and

collect usage data. C9) • The client's operating system begins executing the requested application located remotely on a server. The operating system memory-maps the application and begins executing it, with the application remote file interface code obtaining control whenever the client system's page fault handler determines that the application's page is located on the remote disk drive. The page fault handler asks the application remote file interface code to place the appropriate page data in main memory. The application remote file interface code sends a request to the cache manger for the desired data. (pg. 13, 4th full paragraph). C12) • Data flow description (pg. 6) o The OS loads the application executable into memory and runs the executable. (pg. 7, step 12). o The executable file image is loaded into memory and starts executing. The application files will continuously be loaded into memory as needed. (pg. 7, step 13).	
C3) • Functionality (pg. 1) o The cache manager manages the on-disk cache of file system data, and the in-memory data structures for managing this cache. (pg. 1, para. 3). • Diagram illustrating overall client architecture comprising cache elements. (pg. 8). • Implementation of the cache manager. (pgs. 11 – 17). C4)	e) storing said page segments in a cache on said client.

• **Cache organization (pg. 1)**
  ○ **The cache will be contained in 2 or more files. One file will contain the cache indices, and one or more files will contain the data blocks for cached files.**
 	
C5)
• **Execution of application (pg. 2)**
  ○ **Contact designated App Server using id info, request file system data as necessary. Respond to running application's requests, collect usage data. Cache portions of applications, file system info, & user preference info.**	

IV. Diligence: We diligently constructively reduced the invention to practice on Nov. 6, 2000. Attached, with dates redacted, as Exhibits D1 through D5 (collectively "Exhibit D") are exemplary documents produced between September 26, 2000 and constructive reduction to practice. These documents are in chronological order, and have redacted dates which occurred at irregular intervals but without interruption extending from our conception of the invention to our constructive reduction to practice of the invention. Exhibit D is as follow:

a) D1: Estream 1.0 planning document

b) D2: Estream server component framework low level design

c) D3: Estream set format low level design

d) D4: Estream 1.0 high level design

e) D5: Estream web server load monitoring applet low level design

Exhibit D correlates to the exemplary independent claim 1. These correlations are for the purpose of example only, and not intended to limit the scope of the claims. TABLE 2 provides a rough correlation between Exhibit D and, for example, independent claim 1:

TABLE 2

EXHIBIT D (Examples only)	CLAIM 1

D1) Server group time estimates for implementation.
 	
D2)
• **Functionality (pg. 1)**
  ○ **The Server Component Framework provides a common basis on which server components are implemented. The framework provides a number of services such as common server initialization and configuration, messaging, state management, logging, and error handling. (pg. 1, para. 1).**
 	
D5)
• **Functionality (pg. 1)**
  ○ **One of the requirements for the Estream web server is a facility for monitoring server load. Per this document, this facility will be provided by a graphical load-monitoring applet that will be available for deployment at customer sites as part of the Estream web server installation.**
	(a) A process for intelligent server streaming of conventionally coded streamed application programs across a computer network while concurrently executing said streamed application programs on a client in a computer environment, comprising the steps of:

D1) • **Content** ○ **AppInstallBlk structure time estimate for implementation.**	**(b) downloading an initial portion of a streamed application program on said client wherein said streamed application program comprises page segments and wherein said initial portion of said streamed application remains on said client after terminating execution of said streamed application by said client;**
D1) • **Server group** ○ **App Server time estimate for implementation.**	**(c) providing an application server;**
D1) • **Content** ○ **File access monitor time estimate for implementation.**	**(d) partitioning said streamed application program into said page segments on said application server;**

○ Packager time estimate for implementation. ○ Estream distribution time estimate for implementation. D3) • Functionality (pg. 1) ○ The Estream set is a data set associated with an application suitable for streaming over the network. • Estream set format diagram (pg. 6).	
D4) • App server ○ The application server is there to handle read requests for files accessed by Estream clients. (pg. 10).	(e) wherein said application server streams said page segments to said client upon said client's request;
D4) • Overview ○ A small client 'player" program to allow local execution of applications that reside on the servers. (pg. 1) ○ The user will now see standard shortcuts for subscribed applications, exactly as though the app were installed locally. (pg. 2, 6th box)	(f) wherein the user starts said streamed application program as if said streamed application program were fully installed on said client;
	(g) wherein specific page segments are requested by said client's file system during execution of said streamed application program such that said streamed application program begins execution on said client prior to downloading all of said page segments; and
D1) • Client group ○ Estream cache manager time estimate for implementation. D4) • Client components (pg. 7) ○ ECM: the Estream cache manager. This is the user-space component that handles requests from the Estream File System	e) storing said page segments in a cache on said client.

Driver, and manages the on-disk and in-memory cache of file contents.	

V. **We hereby declare that all statements made herein of our own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, (18 U.S.C. §1001) and that such willful false statements may jeopardize the validity of this application or any patent issued thereon.**

L Shah

Lacky Shah

Date 9/30/2005

I hereby certify that this correspondence is being deposited with the U.S. Postal Service with sufficient postage as First Class Mail in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA , 22313-1450, on:

Date: 9/7/05 By: Susan Wall

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Appl. No.	**:**	**09/784,699**	**Confirmation No.**	**:**	**3149**
Applicant	**:**	**Benitez et al.**	**TC/A.U.**	**:**	**2157**
Filed	**:**	**February 14, 2001**			
Examiner	**:**	**Hussein A. El Chanti**			
Docket No.	**:**	**30126-8009.US01**	**Customer No.**	**:**	**22918**

Declaration of Prior Invention Under 37 C.F.R. § 1.131

Commissioner for Patents
P.O. Box 1450
Alexandria, VA

I. This Declaration establishes invention prior to September 26, 2000.

II. This Declaration is being made by Sameer Panwar, i.e., a named inventor of the above-identified patent application.

III. Conception: Prior to September 26, 2000, we conceived the inventions currently presented in independent claims 1, 16, and 31, of the above-identified patent application. A list of these claims is attached hereto as Exhibit A. Claim 1 is exemplary of an embodiment of the inventions. Exhibit B includes a listing of files related to a product that is representative of the embodiment claimed in the exemplary independent claim 1. Exhibit B includes versions of software and documentation that were created in a Content Management System (CMS) prior to September 26, 2000. The dates of each file have been redacted.

Exhibit C includes a subset of content from the CMS as of September 25, 2000. The content is from files listed in Exhibit B, which are entitled Exhibit C1 through C18. Exhibit C correlates to the exemplary independent claim 1. These

correlations are for the purpose of example only, and not intended to limit the scope of the claims. TABLE 1 provides a rough correlation between Exhibit C and, for example, independent claim 1:

TABLE 1

EXHIBIT C (Examples only)	CLAIM 1
C2) • Functionality (pg 1) o The Client Network Interface (CNI) provides the interface for sending message to servers and provides threads for receiving responses and dispatching them appropriately. (para. 1). • Asynchronous Server Calls (pgs. 5 – 6) o The network send thread is periodically awoken, and it coalesces requests off the NW request queue and sends them to the server (pg. 6, para. 1). o The network receives thread waits for responses to come back from any server. (pg. 6, para. 2). o Finally, the response dispatch thread pulls responses off the response queue, and handles the work of dispatching them appropriately. (pg. 6, para. 3) C6) • Estream client network interface o Handles requests from Estream cache manager o Handles protocol interface to/from server C7) Diagram illustrating structure of server streaming of application programs across a computer network while executing application programs on an Estream client. C17) Abstract and descriptions of CORBA illustrate an implementation of a server framework for Estream. (pg. 1). C18) Diagram illustrating structure of server streaming of applications programs across a computer network while executing application programs on an Estream client.	(a) A process for intelligent server streaming of conventionally coded streamed application programs across a computer network while concurrently executing said streamed application programs on a client in a computer environment, comprising the steps of:

(pg. 7)	
C1) • AIMInstallApplication Prototype (pg. 16) o Step 5. Initializing the profile and prefetch data for this app. C5) • Installation of application o Contact designated App Server using id info, download meta-data about app, potentially including registry/DLL/filesys spoofing info, prefetching info, initial cache contents for app. o Perform initial installation & setup for app, after checking system for previously installed version of app & issuing any appropriate warnings. C10) • Functionality (pg. 1) o The AppInstallBlock is a block of code and data associated with a particular application. This AppInstallBlock contains the information needed by the Estream client to "initialize" the client machine before the Estream application is used for the first time. C11) Data format of the AppInstallBlock.	(b) downloading an initial portion of a streamed application program on said client wherein said streamed application program comprises page segments and wherein said initial portion of said streamed application remains on said client after terminating execution of said streamed application by said client;
C7) Estream client-server diagram comprising an application server. C8) • Finding an application server for a volume. (pg. 2). o The SLM will tell the client which application servers currently provide each volume. It may be necessary for the client to periodically poll the SLM to get up-to-date information about the state of the application servers.	(c) providing an application server;

C9)
• **Technical description of the invention. (pg. 12)**
  ○ **An application file server: Responds to requests by client application cache manager for portions of application's files and directory structure on the server. Transmits compressed information for better bandwidth utilization. (pg. 12, item #4).**
 	
C16)
• **Functionality (pg. 1)**
  ○ **The primary job of the App Server is to service client requests for application data blocks. (pg. 1, para. 4).**
  ○ **The App Server serves data derived from Estream Sets. (pg. 1, para. 5).**
 	
C18)
• **Application Server (pg. 26)**
  ○ **The application server is there to handle read requests for files accessed by Estream clients. Any file accessed on a client through the EFS can have this read request passed to an app server.**	

C7) Estream client-server diagram illustrating transmission of Estream sets.
 	
C9)
• **The process of building a new set of request replies for an application is called building an application stream set. (pg. 15, 3rd full paragraph).**
• **An application stream set contains (pg. 15, 3rd full paragraph):**
  ○ **A unique name of the application for reference purposes,**
  ○ **An index table used to quickly determine which reply to return for a given request,**
  ○ **The set of all possible request**
	(d) partitioning said streamed application program into said page segments on said application server;

replies. • The application stream set is built in the following manner.... (pg. 15, 4th full paragraph). C12) • The Estream Builder is a software program. It is used to convert locally installable applications into a data set suitable for streaming over the network. The streaming-enabled data set is called the Estream Set. This document describes the procedure used to convert locally installable applications into the Estream Set. (pg. 1, para. 1). C13) Estream Builder data flow diagram illustrating the conversion of locally installable applications into Estream sets. C14) • Functionality (pg. 1) o The Estream Application Builder Package Manager is responsible for packaging data gathered from the Installation Monitor, the Profile Manager, and the Upgrade Monitor into a set of data called the Estream set. (pg. 1, para. 1). C15) • Functionality (pg. 1) o The Estream set is a data set associated with an application suitable for streaming over the network. The Estream set is generated by the Estream Builder program. This program converts locally installable applications into the Estream set. This document describes the format of the Estream set. (pg. 1, para. 1). o Diagram illustrating format of the Estream set. (pg. 5).	
C7) Estream client-server diagram illustrating transmission of Estream sets to	(e) wherein said application server streams said page segments to said client upon said

Estream client. **C9)** • **From the point of view of the client and its operating system, the application is resident locally on the client; the execution controller negotiates with an appropriate server to allow the client to obtain (as needed) segments of the associated application files located on the servers. (pg. 12, para. 2).**	**client's request;**
C9) • **The execution controller is given an argument indicating which application is to be executed. From the point of view of the client and its operating system, the application is resident locally on the client; the execution controller negotiates with an appropriate server to allow the client to obtain (as needed) segments of the associated application files located on the servers. (pg. 12, para. 2).** • **If a server accepts the task of serving the application to the client, the execution controller passes the application access request on to the application remote file interface code. This code allows the client to reference file and directory information associated with the remote application as if it resided on a local physical disk device. (pg. 13, 3rd full paragraph).**	**(f) wherein the user starts said streamed application program as if said streamed application program were fully installed on said client;**
C5) • **Execution of application (pg. 2)** o **Send unique certificate for application to appropriate ASP DRM Server, get back id for closest/best App Server & session id.** o **Contact designated App Server using id info, request file system data as necessary. Respond to running application's requests,**	**(g) wherein specific page segments are requested by said client's file system during execution of said streamed application program such that said streamed application program begins execution on said client prior to downloading all of said page segments; and**

collect usage data. C9) • The client's operating system begins executing the requested application located remotely on a server. The operating system memory-maps the application and begins executing it, with the application remote file interface code obtaining control whenever the client system's page fault handler determines that the application's page is located on the remote disk drive. The page fault handler asks the application remote file interface code to place the appropriate page data in main memory. The application remote file interface code sends a request to the cache manger for the desired data. (pg. 13, 4th full paragraph). C12) • Data flow description (pg. 6) o The OS loads the application executable into memory and runs the executable. (pg. 7, step 12). o The executable file image is loaded into memory and starts executing. The application files will continuously be loaded into memory as needed. (pg. 7, step 13).	
C3) • Functionality (pg. 1) o The cache manager manages the on-disk cache of file system data, and the in-memory data structures for managing this cache. (pg. 1, para. 3). • Diagram illustrating overall client architecture comprising cache elements. (pg. 8). • Implementation of the cache manager. (pgs. 11 – 17). C4)	e) storing said page segments in a cache on said client.

• **Cache organization (pg. 1)**
  ○ **The cache will be contained in 2 or more files. One file will contain the cache indices, and one or more files will contain the data blocks for cached files.**
 	
C5)
• **Execution of application (pg. 2)**
  ○ **Contact designated App Server using id info, request file system data as necessary. Respond to running application's requests, collect usage data. Cache portions of applications, file system info, & user preference info.**	

IV. **Diligence:** **We diligently constructively reduced the invention to practice on Nov. 6, 2000. Attached, with dates redacted, as Exhibits D1 through D5 (collectively "Exhibit D") are exemplary documents produced between September 26, 2000 and constructive reduction to practice. These documents are in chronological order, and have redacted dates which occurred at irregular intervals but without interruption extending from our conception of the invention to our constructive reduction to practice of the invention. Exhibit D is as follow:**

a) **D1: Estream 1.0 planning document**

b) **D2: Estream server component framework low level design**

c) **D3: Estream set format low level design**

d) **D4: Estream 1.0 high level design**

e) **D5: Estream web server load monitoring applet low level design**

Exhibit D correlates to the exemplary independent claim 1. These correlations are for the purpose of example only, and not intended to limit the scope of the claims. TABLE 2 provides a rough correlation between Exhibit D and, for example, independent claim 1:

TABLE 2

EXHIBIT D (Examples only)	CLAIM 1
D1) Server group time estimates for implementation. D2) • Functionality (pg. 1) o The Server Component Framework provides a common basis on which server components are implemented. The framework provides a number of services such as common server initialization and configuration, messaging, state management, logging, and error handling. (pg. 1, para. 1). D5) • Functionality (pg. 1) o One of the requirements for the Estream web server is a facility for monitoring server load. Per this document, this facility will be provided by a graphical load-monitoring applet that will be available for deployment at customer sites as part of the Estream web server installation.	(a) A process for intelligent server streaming of conventionally coded streamed application programs across a computer network while concurrently executing said streamed application programs on a client in a computer environment, comprising the steps of:
D1) • Content o AppInstallBlk structure time estimate for implementation.	(b) downloading an initial portion of a streamed application program on said client wherein said streamed application program comprises page segments and wherein said initial portion of said streamed application remains on said client after terminating execution of said streamed application by said client;
D1) • Server group o App Server time estimate for implementation.	(c) providing an application server;
D1) • Content o File access monitor time estimate for implementation.	(d) partitioning said streamed application program into said page segments on said application server;

○ Packager time estimate for implementation. ○ Estream distribution time estimate for implementation. D3) • Functionality (pg. 1) ○ The Estream set is a data set associated with an application suitable for streaming over the network. • Estream set format diagram (pg. 6).	
D4) • App server ○ The application server is there to handle read requests for files accessed by Estream clients. (pg. 10).	(e) wherein said application server streams said page segments to said client upon said client's request;
D4) • Overview ○ A small client 'player" program to allow local execution of applications that reside on the servers. (pg. 1) ○ The user will now see standard shortcuts for subscribed applications, exactly as though the app were installed locally. (pg. 2, 6th box)	(f) wherein the user starts said streamed application program as if said streamed application program were fully installed on said client;
	(g) wherein specific page segments are requested by said client's file system during execution of said streamed application program such that said streamed application program begins execution on said client prior to downloading all of said page segments; and
D1) • Client group ○ Estream cache manager time estimate for implementation. D4) • Client components (pg. 7) ○ ECM: the Estream cache manager. This is the user-space component that handles requests from the Estream File System	e) storing said page segments in a cache on said client.

Driver, and manages the on-disk and in-memory cache of file contents.	

V. We hereby declare that all statements made herein of our own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, (18 U.S.C. §1001) and that such willful false statements may jeopardize the validity of this application or any patent issued thereon.

Sameer Panwar

Date 8/25/2005